Cross-Layer Self Routing: a self-managed routing approach for MANETs

M.A. Razzaque, Simon Dobson and Paddy Nixon
Systems Research Group
School of Computer Science and Informatics
UCD Dublin IE
Email: abdur.razzaque@ucd.ie

***Abstract*—Mobile *ad hoc* networks (MANETs) generally adopt a peer-to-peer architecture in which the nodes themselves provide routing and services to the network. Disconnectivity with peer nodes, induced by mobility, power drains and damage makes route maintenance difficult and degrade the network's ability to offer services reliably to its peer nodes. In this paper, we present a routing scheme for proactive management of such disconnections, by fusing and leveraging information derived from multiple levels of the network protocol stack using cross-layering. In addition to the disconnectivity information, this routing scheme utilises node's service level information and data/service replication to provide service from an alternate source (if there is one) even in the absence of the targeted source. Simulation results demonstrate significant improvements in route maintenance and service availability over other similar schemes.**

I. Introduction

A key challenge in MANETs is to work out efficient methods to ensure route availability while incurring minimal control overhead. Routing techniques may be broadly divided into proactive and reactive schemes [1]. Proactive protocols often suffer from excessive control overhead associated with maintaining routes to destinations even when not required; reactive protocols experience higher end-to-end packet delays compared to proactive protocols, since routes must be re-formed during communication. Hybrid protocols like the Zone Routing Protocol (ZRP) offer the qualities of both proactive and reactive protocols. The most desirable routing protocol is one which offers minimal end-to-end packet delays for real-time traffic and less control overhead for non-real-time traffic. This can only be achieved by exploiting link state information (such as link life time) that is generally ignored in conventional *ad hoc* routing protocols: even the existing hybrid protocols do not utilise link state information, and hence do not offer enormous performance advantage over existing reactive or proactive protocols.

Link Life Time (LLT) is one of the most important link state information, mainly controlled by disconnectivity due to mobility and the node failure due to power shortage. If we can exploit the power- and mobility-related information in routing, there is an opportunity to provide better route maintenance and improve the route performances– using link-layer information to condition transport- and network-layer behaviour. Existing modular or layered approaches would not allow such cross-layer interactions, so a cross-layer architecture may provide a better basis on which to build [2], [3].

The LLT of a wireless link can be predicted exploiting location and movement pattern information and/or remaining power information in the node using a cross-layer approach. Predicted LLT (LLT_p) can be utilised for proactive route maintenance and the corresponding route could maintain service only if the providing node is within the coverage area and not dead. But by exploiting a node's service level information and data/service replication we might be able to provide service from an alternate source (if there is one) even in the absence of the targeted source. To our knowledge no single paper has considered all these issues for a MANET routing protocol. Our present work does not start from scratch rather builds on an existing reactive *ad hoc* routing protocol, AODV, a widely studied routing protocol in MANET environments. We modify the AODV's route maintenance and route discovery algorithms to exploit cross-layer information concerning movement and power to define a new protocol, Cross-Layer Self Routing (CLSR).

The remainder of this paper is structured as follows. Section II briefly describes cross-layer networking. A brief related study is presented in III . Section IV presents CLSR in detail, along with the LLT_p calculation, and service-level information, data replication and their corresponding uses in CLSR. Section V describes the implementation and evaluation. Section VI concludes the work with some future directions.

II. Cross-Layering

To exploit the link state information like LLT, we need to capture the interactions between the physical, link and network layers – but existing strict layering schemes do not support such interactions. For the exploitation of service/data level information in routing, the network layer has to interact with the application layer, but again a strictly layered approach does not allow interactions amongst non-adjacent layers.

Recent research studies [3], [2] show that the cross-layer design principle has great importance in wireless *ad hoc* networks, where different layers are more likely to use the *same* information in making layer-specific decisions. In particular, the real-world locations of the nodes and the topology of the network are commonly used by both the routing and the application layers in computing routes and making higher-level decisions. Such redundancy complicates the design and magnifies the possible impact of uncertainty.

To support cross-layering in a principled way we need a cross-layer architecture. A number of cross-layer architectures have been published in the literature: for our purposes we will consider the architecture developed in [4] as it supports the formation of local and global views of network information within a managed and data dissemination framework.

III. Related work

Exploitation of cross-layering in wireless routing has demonstrated considerable potential. Some authors have exploited cross-layering for routing in wireless networks. In [5] a cross-layer approach has been used to exploit mobility information to enhance the performance of AODV, but considers only constant transmission power which is not true most of the cases. Cross Layer AODV [6] is an efficient routing protocol crossing the routing and MAC sub-layers. It is based on standard AODV routing protocol and utilises useful information of MAC sub-layer in routing but does not explicitly utilise the remaining node power or location information. A way to improve data accessibility service for a group of mobile users to access desired data has been presented in [7]. To do that it utilises cross-layer assisted predictive location-based QoS routing protocol as well as the replication services. Authors in [8] exploit three primitive physical layer parameters: interference, packet success rate, and data rate in cross-layer based routing, whereas [9] considers cross layer interaction between routing and MAC layer for their directional routing and directional neighbour tables based on demand routing scheme. None of the above works exploits the information related to mobility, power failure and service together in routing for MANETs.

IV. CLSR

A MANET's self-configuring and self-organising properties reflect through its dynamic topology. The routing protocol in MANETs adapts the topology to the physically possible communication links. In this sense every MANET routing protocol somehow shows the self-organising properties. Route discovery and route maintenance are the main functional entities of a routing protocol, and in the following sections we describe the modifications we have made to the underlying AODV protocol's route discovery and the route maintenance mechanisms in order to develop CLSR. Before describing the modifications, we briefly discuss the link life prediction, cross-layer service discovery and data/service replication, which are key for the modifications.

A. Link life time prediction

Link Life Time (LLT) is the time for which a link between nodes exists. A link could be broken if one of the nodes goes out of transmission range or dies. LLT can be predicted – to a certain extent – using mobility-induced and node failure induced disconnectivity information.

1) LLT_p based on mobility information: LLT_p of a link between two mobile nodes can be predicted based on the distance between them and their relative velocity. In turn, this predicted lifetime can be used for the proactive route maintenance. Our LLT_p will be based on the one developed in [5], with some modifications. These authors assume constant transmitter power, but in MANETs power control mechanism which uses variable transmission power based on distance between the nodes is readily available. So, we will use the variable transmission power of a node which is controlled by the transmission power control algorithm in a manner similar to the approach of [10], [11].

There is a relation between the transmitted signal power, received signal power and the distance between the two nodes. We can exploit their relation to calculate the distance. The distance d between a transmitting antenna and an observing point can be easily computed if the received signal strength P_r and the respective radio propagation model are known. In the Two-Ray Ground Reflection model [12] P_t (transmitted signal power) and P_r are related as

$$P_r = \frac{k.P_t}{d^4} \tag{1}$$

where k is the constant depends on the antenna height and gain. There are basically two ways to predict the connectivity between two neighbouring nodes. The first method assumes knowledge of motion parameters of the two neighbours (speed, direction, and transmission range), from which the LLT_p of their link can be determined. The second method uses received signal power measurements to predict LLT_p. This method has been proposed in [13] and assumes that the sender power level is constant. Received signal power levels are measured from the packets received from a mobile node's neighbour. Substituting this received power(P_r) and the sender power(P_t) in equation (1), the distance of separation between the two nodes can be calculated. Finally, comparing this distance with the transmission range (R), nodes can predict when they will move out of transmission range of each other. This algorithm estimates the velocity of the node based on the radial distance that the node has traveled and the time elapsed since the last observation. The estimate is derived from the change in signal strength of the received MAC frames. From the computed value of velocity, the algorithm conservatively estimates the time when the concerned link will break. This algorithm is very heuristic in nature (for details see [13], [5]).

The predicted value of LLT_p will be used in CLSR for the proactive route maintenance procedures. Unfortunately, the prediction algorithm may not be very accurate as the nodes keep changing their speed and direction randomly. The accuracy of the prediction algorithm increases as the rate of the number of packets received increases.

2) Node's remaining power information and LLT_p: Node failure due to power shortage is another significant cause of route failure, so by predicting the remaining power in a node, we can predict the possible node failure if the remaining power goes below some threshold value. This predictability on node failure can be exploited in CLSR for the intelligent route maintenance procedures. Power failure implies $LLT_p = 0$, which requires immediate remedial action.

We can predict the remaining power either by direct measurement or calculating the power consumptions and subtracting it from the total initial power. Calculating power consumptions in a mobile node is not a trivial task, since most nodes will employ quite complex power management strategies to increase battery life. Some authors (for example [14], [15]) present details of power consumption, but all focus on the power consumed by the communication device (typically a wireless network card). These methods of power calculation are quite useful in power-aware routing but in our case we need to derive accurate estimates of the remaining power. On the other hand, non-communication-related power consumption varies according to what the node does – including the power calculations themselves. So, estimating the remaining power of a mobile node with an acceptable accuracy is a difficult task, and we prefer to use the direct measuring method using a simple analogue-to-digital converter standard on all microcontrollers (and even some general computing devices). In the cross-layer architecture of [4] the knowledge plane maintains a view of the current battery status. If the battery voltage drops below a preset value, this triggers a power warning event, hence disseminate this event to rest of the nodes to form a global view in this event's respect. Finally, CLSR exploits this event information in routing.

B. *Cross-layer service discovery, data/service replication*

Conventional service discovery mechanisms ([16], [17], [18], etc) have limited knowledge of network topology and assume a mostly static environment with infrequent topology changes. In contrast, frequent topology changes are the norm in MANETs, and good service selection is highly dependent on up-to date knowledge of the network topology. By performing service discovery in the same way as route discovery, nodes can accumulate the routing information while performing the service discovery and disclose service level information at the routing layer, which will improve the overall performances [19], [20] in MANETs like environments. Therefore, CLSR incorporates the options for cross-layer integrated service discovery.

Using figures 1 and 2 we can show two possible ways of this service level information exploitation in routing. (i) In figure 1(a) s is our concern node and $d1$ and $d2$ are two servers. Node s node has a request for x type of service, and s knows (through route level service information & global view) that both $d1$ and $d2$ can meet the request and it uses the route s-$n1$-$n2$-$d1$. Similarly, $n2$ knows that both $d1$ and $d2$ could provide the x type service. If during the transmission $d1$ fails, existing routing protocols would have $n2$ receiving the packet, determining $d1$ to be dead and finally sending a "node unreachable" error message to s, which wastes all the resources committed to the exchange. Use of CLSR leads to scenario (b) where nodes have re-organised because of the death of $d1$, and once $n2$ gets the request from s it reroutes to $d2$ instead of $d1$ and meet the request. (ii) In figure 2(a) $d1$ moves out of the transmission range and become unreachable and $d2$ provides the service and later on $d2$ fails due to power shortage. Using existing routing schemes with strict layering there will be no service available for s. Using CLSR with data/service replication we may provide the service to s. CLSR will attempt data/service replication if it knows that within the subnet/network there is only one server for a particular service and it's going out of range or dying. In CLSR the remaining power information gives a power warning before the complete dead of a node, so $d2$ gets the warning before its death. If the service replication (say to node $n2$) is successful then s gets service from $n2$. Service replication is limited to services that are not tight to pre-installed infrastructure or are machine dependent. Both of these approaches can conserve energy and minimise latency by eliminating the overhead required to invalidate the current route, establish a new route, and retransmit the request. We use the similar approach used in [7] for data/service replication.

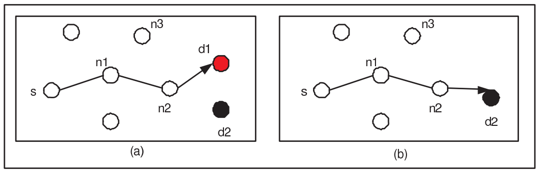


Fig. 1. Sample network scenario one

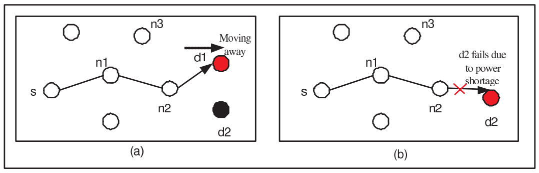


Fig. 2. Sample network scenario two

C. *Route discovery in CLSR*

CLSR follows almost the same route discovery method as AODV with two different situational modifications. One is for the exploitation of service level and another for remaining power information. For the exploitation of service level information, CLSR supports cross layer service discovery (CLSD) and to do that it introduces two extra message types, i.e. $ServiceRequest(SREQ)$ and $ServiceReply(SREP)$. To handle these messages, we did the necessary modifications to route discovery and route maintenance. During the route discovery, inclusion of a node with power warning or a very little power is inefficient as that node may die soon and make the route invalid even before the route replies. On receiving the $SREQ$, a node checks the power warning, if it is true does nothing otherwise appends itself to the source route and verifies if it hosts a service that matches the service description. If so, the node replies to the source by sending a $SREP$ *via* a reversed source route; otherwise, the node rebroadcasts the $SREQ$. Finally, the node that initiated the service discovery finds out the identity of one or more nodes that host services matching the service description and one or more source routes to these nodes. If a node gets a power warning during forwarding a $SREP$, it sends an error message to the originator of $SREQ$ and originator takes further actions. Like AODV, CLSR reduces the latency and frequency of service discoveries by allowing intermediate nodes to cache overheard mappings between service descriptions and service locations, and to respond to service requests for which they have a cached mapping. CLSR learns about topology changes implicitly when either a source route breaks or it overhears a service reply that has a new service mapping. In CLSR, a node gets the power warning from the physical layer based on the remaining power information and then using the global view formation scheme of the cross-layer architecture [4] disseminate this information to the network. Thus most of the nodes get the information and can exploit it when needed.

D. Route maintenance in CLSR

Route maintenance is the key contribution of the CLSR which utilises both the link life time as well as the remaining power initiated power warning. In addition to them, if CLSD option is activated, CLSR exploits the service level information to maintain a proper route to the server and satisfy the user's request. After the successful route discovery, at the network layer of each node, each link connecting neighbouring nodes is periodically monitored for possible breakage in the near future. Usually, only active links connecting nodes those are moving outward direction (with respect to the concerned node) or those nodes that might have power warning are of particular interest, because they are considered to be vulnerable candidates for link breakage.

In AODV a route may be in any one of the three states:

- UP route still exists; packets forwarded only if route is in this state.
- UNDER_REPAIR route is being locally repaired.
- DOWN route is broken; used mainly to flush routes out of the routing table.

In CLSR a route can additionally be in a fourth state:

- PROACTIVE_REPAIR packets can be forwarded using this route, but the route is currently under proactive repair.

A link that is about to break (say within MIN_THRESHOLD: 0.03sec) will render all the routes that use this link invalid. But it is not necessary to proactively re-discover *all* routes that make use of the broken link; instead routes are proactively re-discovered only for active routes. When CLSR switches to this state, it initiates a local route repair mechanism for all active routes using the neighbour in question as the next hop, if the upstream node is closer to the destination than to the source. Otherwise, link breakage is allowed to happen, and normal AODV like route error handling mechanisms take over. The routes (other than the vulnerable one) discovered during a proactive repair are cached in the routing entries for those particular destinations with their corresponding expiry times. In case of multiple route replies, the selection criteria for a route are the same as that for a route discovered during normal AODV route discovery mechanism [21], [5].

If indeed a link breaks before the cached route expires, the existing routing table entries that make use of the broken link are replaced with the cached routes. In the event that the link breaks in the absence of cached routes, normal AODV route error handling procedures for those routes are initiated. Link breakage is determined in one of the following ways:

- Periodically LLT_p is checked for each link and for any link, if LLT_p has elapsed, then the link is assumed to be broken and routes are replaced as described above. Success of this method depends on the accuracy of the prediction algorithm mentioned earlier.
- If LLT_p is predicted (erroneously) to be later than the actual link breakage time, the link breakage can be discovered using link layer acknowledgements, if attempts were made to route a packet over the broken link. In this situation, the unexpired route in the route cache, if present, is used to replace the broken route. Otherwise, a new route discovery is initiated. This method of determining link breakage may be needed mainly if the link has been idle for a long time since the last predicted value of LLT_p. Generally, this method is quite time consuming, because the link layer can determine that a link is broken only after a series of retransmissions, which are initiated when acknowledgements are not received from the node at the other end of the link.
- Periodically checks the remaining power of the nodes in each link and if it has become less then the threshold P_{th}, generate a power warning and this causes link breakage in the near future. If the node with power warning is a forwarding node then it forwards existing packet (if any) and asks for immediate proactive route maintenance. If any route found, existing route replaced by that. If it is the destination node, it sends a power error message to the immediate previous node which will select another available server node which can provide the same service; if there is no available server node for this service send error message to the source and ask for necessary actions. Finally, if it is the only server node than it might attempt data/service replication if the situation allows.

V. Implementation and evaluation

For the experimental implementation and evaluation of CLSR we have constructed a simulation based on ns-2 [22]. For cross-layer interaction and the dissemination of the power warning, we will use the architecture presented in [4]. The knowledge plane is the key element of that architecture through which cross-layer interactions occurs between different non-adjacent layers. "Global view" of the knowledge plane is responsible for the dissemination of power warning like global information. Mobility-induced disconnectivity information is contributed by the cross-layering among the network, MAC/link and physical layers, whereas cross-layering between the physical and network layers contributes remaining-power

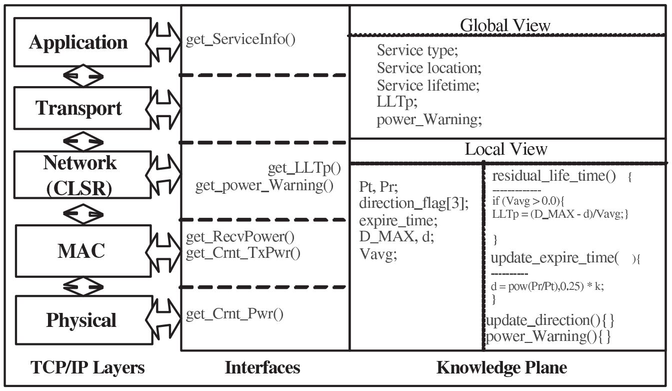


Fig. 3. A snapshot of the implementation

information. For CLSD, we do the cross-layering between the network and application layers. Due to space limitation, we describe only the key parts of the implementations: we refer interested readers to our other work for further details.

The MAC layer's $recv()$ function receives any frames from the wireless physical layer destined to that particular node or any upper layer packets destined to other nodes. The MAC frame header contains fields for received packet's power P_r and the transmission power P_t at which this packet is transmitted. Hence probing the incoming MAC frame headers, both the parameters of incoming packets are extracted. Based on these parameters, the radial distance d between the receiving node and the sending node is computed using the equation (1). This value of d is fed to the prediction algorithm, which predicts the time of link breakage and this link lifetime with their corresponding information forms a table. As the table with node-link lifetime information has to be accessible to both the MAC layer and the CLSR, we placed it in the common knowledge plane of the cross-layer architecture as shown in figure 3. The snapshot of the implementations in figure 3 shows $residual\_life\_time()$, $update\_expiretime()$, $update\_direction()$ and $power\_Warning()$ are the key functions in the implementations, which calculate the LLT_p, the expiry time of a link and d, node's directions and power warning respectively.

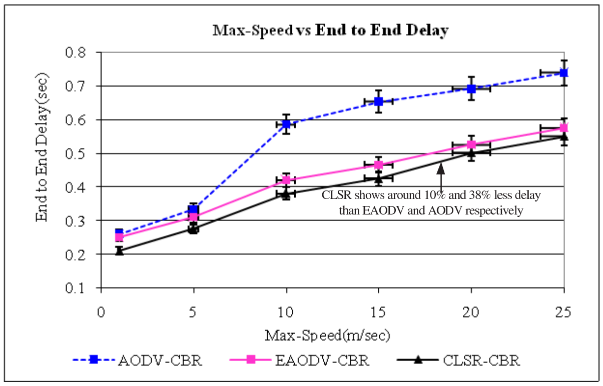
For the node's power warning related information, (again shown figure 3, the knowledge plane periodically collects the current node's power through $get\_Crnt\_Pwr()$ from the physical layer and compares it with the threshold (P_{th}) power. If the remaining power is less than P_{th}, the knowledge plane generates a power warning and pass it to "global view" for the dissemination. To make service-level information available to routing layer or CLSR, we implement the CLSD approach similar to [20].

Through the aforementioned approaches LLT_p, remaining power and service-level information are available to be exploited in CLSR. At the CLSR layer at each node, each link connecting neighbour nodes is periodically (once every 0.5 seconds) monitored for possible breakage or power warning in the near future. Typically, active links connecting nodes that are moving outward are of particular interest, because they are considered to be susceptible candidates for breakage. If there is any link breakage or power warning CLSR will go for proactive route repair or any other necessary action as mentioned earlier. If extreme situation like figure 2 happens, it will go for data/service replication using service level information. For data/service replication we are assuming that all the nodes are capable of receiving replicated data or service.

We performed simulations with the AODV (LLACKS-enabled version), EAODV [5] and CLSR to compare performance metrics of the protocols. For space reasons, we report results on two key metrics: end-to-end delay and control overhead ratio (the ratio of total control overhead measured in bits to the total data bits transmitted successfully). For simulation, we consider both CBR (Constant Bit Rate) and TCP traffic and for mobility modelling we use the Random Waypoint (RW) Model. The simulations using RW model were run in a 1500m by 1500m area with 50 nodes under varying conditions of mobility. In the graphs bars around each point indicate 95% confidence interval.

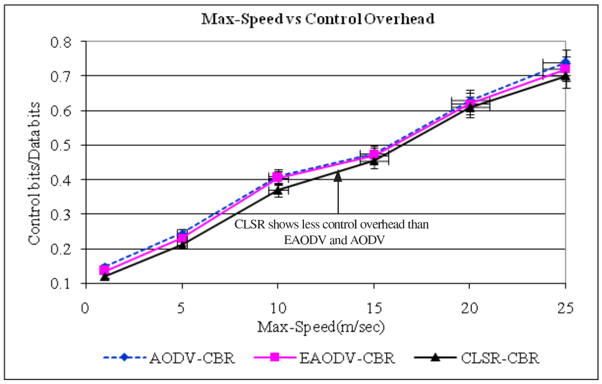
Figure 4 and 5 present the results for end-to-end delay and control overhead of CBR traffic with respect to maximum node velocity/speed respectively. As shown in figure 4, for very similar packet delivery ratio EAODV offers lesser end-to-end delay than AODV, whereas CLSR offers lesser delay than both AODV and EAODV with increase in maximum node velocity/speed. At higher speeds, CLSR shows around 38% and 10% lower end-to-end delay than the AODV and EAODV respectively. In similar communication pattern, an increase in maximum velocity will increase the rate of change of topology, which will reduce the average lifetime of a link. This in turn will increase the number of RREQs(Route Requests), which will increase the control data (and hence the network traffic) transmitted. An increase in network traffic implies an increase in the rate of packets (samples) fed to the link-layer prediction algorithm, which increases the accuracy of link breakage predictions. And this is why both EODV and CLSR show lesser delay. AODV and EAODV deal the node failure related link breakage reactively, whereas CLSR does it proactively, which helps it in advance route discoveries and thus shows lesser delay. Even use of Transmission Power Control (TPC) [10] in CLSR could reduce delays and power consumption. The increased network control traffic (high priority) causes the increase in queuing delay of data (low-priority) packets. This is shown in figure 4: an increase in velocity/speed increases the end-to-end delay of packets.

Figure 5 shows the variations in control overhead ratio with respect to maximum velocity/speed of the nodes. For very similar packet delivery ratio, CLSR outperforms AODV and EAODV with respect to control overhead. This is contributed by the reactive route repair (initiated by link breakage or power warning), which uses local route repair instead of globally through out the network. This requires lesser RREQs if it is successful and ultimately reduce overhead. If it fails the situation can be reverse. Even, unwanted proactive route discoveries in EAODV and CLSR could reverse the situation.



CLSR shows around 10% and 38% less delay than EAODV and AODV respectively

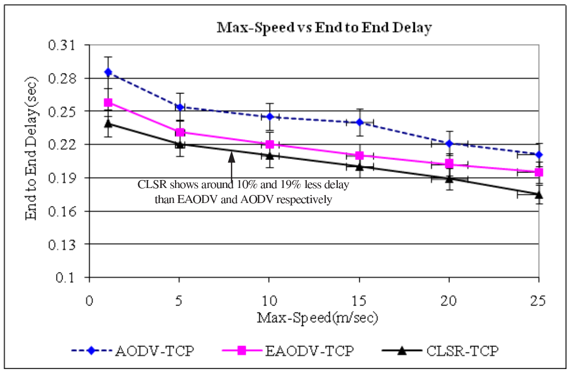
Fig. 4. CBR traffic: end-to-end delay



CLSR shows less control overhead than EAODV and AODV

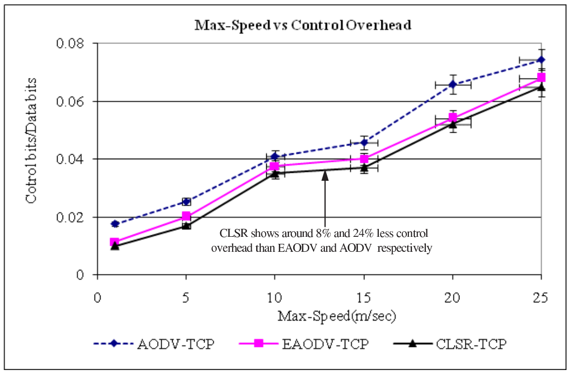
Fig. 5. CBR traffic: control overhead

Therefore, attention is required in prediction algorithm to keep the CLSR's overhead lesser.



CLSR shows around 10% and 19% less delay than EAODV and AODV respectively

Fig. 6. TCP traffic: end-to-end delay



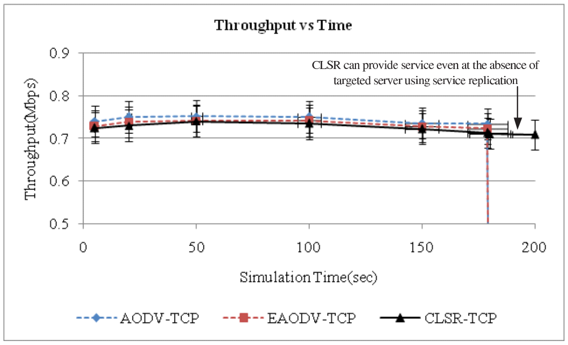
CLSR shows around 8% and 24% less control overhead than EAODV and AODV respectively

Fig. 7. TCP traffic: control overhead

Results for end-to-end delay and control overhead ratio of TCP traffic with respect to maximum node velocity/speed are presented in figures 6 and 7 respectively. Result in figure 6 shows that for very similar throughput, CLSR offers around 19% and 10% less end-to-end delay than the AODV and EAODV respectively. The reasons for this improvement in end-to-end delay are similar to the mentioned earlier for CBR traffic end-to-end delay. The trend in figure 6 shows that end-to-end delay decreases with increasing velocities. This is because with increasing velocities, routes are broken quickly and also made quickly, while at lower velocities, routes are broken slowly and also made slowly. At lower velocities, once the route is broken, retransmissions may occur because the delay in forming a new route is higher. The drop off in delay due to higher rate of bursts possible at lower velocities is offset by the increase in delay due to retransmissions as a result of broken links and increase in queuing delay due to congestion (and possible retransmission) at higher source rates. The outcome of increased queuing delay due to increased network traffic due to numerous route discoveries at higher velocities is lesser than the effect of increased delay due to congestions and retransmissions in the lower velocity case, and hence end-to-end decreases with increasing velocities.

Control overhead is one of the key parameters in TCP as end-to-end delay in CBR traffic. Figure 7 shows this control overhead related performance with respect to variation in maximum velocity for TCP traffic. For almost similar throughput, CLSR offers around 24% and 8% lower control overhead than the ADOV and EAODV respectively. The reason for decrease in control overhead in CLSR could be credited to the outstanding performance of the prediction algorithm and power warning scheme. In TCP traffic, the number of MAC frames carrying IP-encapsulated TCP segments is comparable to the number of control packets generated, and hence the prediction algorithm has the luxury of predicting link breakage time with the help of a large number of sampling packets. As a result, with CLSR and EAODV, the number of link breaks in active routes is reduced, when compared to the number of link breaks in AODV, which reduces the control traffic generated.

Figure 8 shows the performance of CLSR, when used in an extreme situation like in figure 2. For this case we are considering the scenario: there are two nodes in our 50 nodes network, which provide x-type service (for example a map viewer), and sometime around 150s one of these two nodes moved out range and some time later, other node dies for power shortage. When the first node moves out of range; CLSR, AODV and EAODV can still support x-type service through the remaining node. As CLSR uses service level information, so it provides the service with lesser delay and



CLSR can provide service even at the absence of targeted server using service replication

Fig. 8. TCP throughput with service replication

control overhead ratio than EAODV and AODV. As all the nodes (after the move out of one server node) are getting x-type service from the remaining node, so due to congestion throughput decreases little bit for all as shown in figure 8. After that, remaining x type service providing node over hit by the service requesters and it drops power and finally dies. As CLSR supports service/data replications alongside the service level information and remaining power based warning, therefore it can still provide x-type service but AODV and EAODV cannot as shown in figure 8.

VI. Conclusion

Dynamic topology changes and scarcity of battery power may cause frequent link breakage in MANETs. This link breakage makes the route maintenance very complex, even degrade performances. By exploiting link state information as has been collected across the entire system (such as link-life prediction, service type and possibly mobility information) through cross-layering across the protocol stack, we improve the route maintenance and increase the possibility of successful service delivery. Utilising information on nodes' service level, capability and so on, and performing data/service replication, we are able provide service from an alternate source (if there is one) even in the absence of targeted source. CLSR, which exploits all this information outperform some similar routing schemes like AODV and EAODV. This performance improvement in MANETs ultimately reduces the service disruptions.

Use of cross-layering approach is not straight forward and unbridled cross-layering could raise loops between the layers and could deliver opposite results. The use of a knowledge plane architecture helps to reduce these risks, but further work is needed to develop an appropriate methodology for cross-layer programming.

We have considered only a small sub-set of potential cross-layer interactions, with a single service class. Supporting multiple service classes simultaneously could, we believe, be greatly advantaged by taking account of (for example) the relative importance of the different service classes (extracted from the application), but doing so also requires capturing and addressing some quite complex trade-offs between services. We intend to address this issue in depth in the future.

Acknowledgment

This work is partially supported by Science Foundation Ireland under grant number 04/RPI/1544, "Secure and Predictable Pervasive Computing" and Higher Education Authority PRTLI4 under grant number R10891, "NEMBES: Networked Embedded Systems."

References

[1] E. D. Mehran Abolhasan a, Tadeusz Wysocki a, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks*, vol. 2, pp. 1–22, 2004.

[2] M. Razzaque, S. Dobson, and P. Nixon, "Cross-layer architectures for autonomic communications," *Journal of Network and Systems Management*, vol. 15, no. 1, pp. 13–27, March 2007.

[3] V. Srivastava and Motani, "Cross-layer design: a survey and the road ahead," *Communications Magazine, IEEE*, vol. 43, no. 12, pp. 112–119, 2005.

[4] M. Razzaque, S. Dobson, and P. Nixon, "A cross-layer architecture for autonomic communications," in *Autonomic Networking*, ser. LNCS, vol. 4195. Springer-Verlag, 2006, pp. 25–35.

[5] P. Mani and D. Petr, "Development and performance characterization of enhanced AODV routing for CBR and TCP traffic," in *Proceedings of the 2004 IEEE Wireless Telecommunications Symposium*, 2004.

[6] Z. R. J. S. W. Gu, "A cross-layer AODV routing protocol," in *Mechatronics and Automation, IEEE International Conference*, vol. 4, August 2005.

[7] K. Chen, S. Shah, and K. Nahrstedt, "Cross-layer design for data accessibility in mobile ad hoc networks," in *In Proc. of 5th World multiconference on systemics, cybernetics and informatics*, 2001.

[8] L. Iannone, R. Khalili, K. Salamatian, and S. Fdida, "Cross-layer routing in wireless mesh networks," in *Wireless Communication Systems, 1st International Symposium on*, 2004, pp. 319–323.

[9] T. C. C. A. D. Gossain, H.; Joshi, "A cross-layer approach for designing directional routing protocol in manets," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 4, 2005, pp. 1438–1541.

[10] M. Razzaque, P. Nixon, and S. Dobson, "Demonstrating the feasibility of an autonomic communications-targeted cross-layer architecture," in *Proceedings of the 14th International Conference on Advanced Computing and Communications*, 2006.

[11] V. Kawadia and P. Kumar, "Principles and protocols for power control in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23(1), pp. 76–88, 2005. [Online]. Available: citeseer.ist.psu.edu/kawadia05principles.html

[12] T. S. et al.., "A survey of various propagation models for mobile communication," *IEEE Antennas and Propagation Magazine*, vol. 45, pp. 51–82, 2003.

[13] P. A. B. Narendran and D. K. Anvekar, "Minimizing cellular handover failures without channel utilization loss," in *Proceedings of IEEE Global Communications Conference*, vol. 3, 1994, pp. 1679–1685.

[14] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *IEEE INFOCOM*, 2001.

[15] J. Ebert, B. Burns, and A. Wolisz, "A trace-based approach for determining the energy consumption of a wlan network interface," in *In Proc. of European Wireless*, 2002, pp. 230–236.

[16] "Bluetooth specification part e. service discovery protocol(sdp)," http://www.bluetooth.com, 1999.

[17] "Jini javaspaces service specification," http://www.sun.com/jini/specs.

[18] S. Czerwinski, B. Zhao, T. Hodes, A. Joseph, and R. Katz, "An architecture for a secure service discovery service," in *In Proc. of MobiCom*, 1999.

[19] A. Varshavsky, B. Reid, and E. de Lara, "A cross-layer approach to service discovery and selection in manets," in *The Second International Conference on Mobile Ad-Hoc and Sensor Systems*, November 2005.

[20] G. P. Halkes, A. Baggio, and K. Langendoen, "A simulation study of integrated service discovery," in *EuroSSC*, 2006, pp. 39–53.

[21] "Rfc(3561) for AODV," http://rfc.dotsrc.org/rfc/rfc3561.html.

[22] "The network simulator," http://www.isi.edu/nsnam/ns/ns-build.html.